A Hybrid Approach to Population Construction For Agricultural Agent-Based Simulation

preprint version: forthcoming in eScience 2016

Peng Chen*, Tom Evans†, Michael Frisby‡ Eduardo Izquierdo* and Beth Plale*
*School of Informatics and Computing, †Department of Geography, ‡Indiana Statistical Consulting Center
Indiana University Bloomington
Email: chenpeng, evans, mfrisby, edizquie, plale@indiana.edu

***Abstract*—An Agent Based Model (ABM) is a powerful tool for its ability to represent heterogeneous agents which through their interactions can reveal emergent phenomena. For this to occur though, the set of agents in an ABM has to accurately model a real world population to reflect its heterogeneity. But when studying human behavior in less well developed settings, the availability of the real population data can be limited, making it impossible to create agents directly from the real population. In this paper, we propose a hybrid method to deal with this data scarcity: we first use the available real population data as the baseline to preserve the true heterogeneity, and fill in the missing characteristics based on survey and remote sensing datasets; then for the remaining undetermined agent characteristics, we use the Microbial Genetic Algorithm to search for a set of values that can optimize the *replicative validity* of the model to match data observed from real world. We apply our method to the creation of a synthetic population of household agents for the simulation of agricultural decision making processes in rural Zambia. The result shows that the synthetic population created from the farmer register can correctly reflect the marginal distributions and the randomness of survey data; and can minimize the difference between the distribution of simulated yield and that of the observed yield in Post Harvest Survey (PHS).**

I. Introduction

Spatial agent-based modeling (ABM) has been proven to be beneficial to agricultural economics for its ability to represent interactions amongst heterogeneous actors, and to fully take into account the spatial dimension of agricultural activities [1]. While an actor (agent) can be an individual farmer, it is typical to consider the household as the basic unit of analysis in agricultural modeling [2], [3], [4]. An agent representing a household has characteristic variables (e.g. wealth, labor supply, preferences) and spatial locations (cells/patches). The key to good agricultural agent based modeling is to construct agents that can truly reflect the characteristics of a real population of households.

However, data about populations is often limited to census data, and in some cases administrative (farmer) register data, which refers to information collected at the district level about which farmers are planting which crops and in which fields in a growing season. This information generally contains a limited set of characteristic variables and thus is insufficient for creating the agents. One way to deal with this insufficient information (i.e., missing agent characteristic variables) is to integrate multiple available data sources and to derive from empirical data. A valuable practice is to use the available census data (and in some cases aggregated administrative register data), combined with remote sensing data, and project that onto the creation of a synthetic population dataset, which is then used to construct agents [3], [4], [5], [6]. While this approach may allow heterogeneities between subgroups of the population to occur, it cannot capture the heterogeneities at individual (household) level. For instance, preserving the structure (sizes) and heterogeneities of subgroups results in synthetic data that may be statistically equivalent to real population (census or aggregated register data) [2], but still lack the household level granularity needed for agent-based simulations.

In this paper, we propose a novel method of simulating synthetic population data based on available real population data (i.e., individual level farmer register data), household survey data, and remote sensing data. The real population data serves as the baseline for the synthetic population. The variability needed within the missing variables, and the relationships between missing variables and known variables are both learned from the survey data, and then used to simulate the missing variables. To assign spatial locations to the synthetic population, the agricultural land data generated from remote sensing is disaggregated into raster form and allocated to the synthetic population.

While researchers in Geography often want to derive as many variables as possible from empirical data, there could always be variables needed that has no relevant real population data or survey data available. In other words, our proposed data simulation method can not derive variables that do not exist in the survey data. In these cases, it is possible to use an optimization process to derive a set of values for the missing variables that improve the *replicative validity* [7] of the model; that is, aiming to minimize the difference between the data generated by the simulation and data previously acquired from the real system. Nevertheless, classical optimization tools such as regression may not be effective in finding a suitable combination of missing variables due to the inherent complexity of the interactions within the model [8]. For this reason, Genetic Algorithms (GAs) have been previously used for model calibration [9], [8], [10], [11] with good results.

In this paper, we apply the microbial genetic algorithm [12]

to the calibration of missing variables of household agents, and demonstrate that it can be used in conjunction with the data simulation method to create a synthetic population of agents. There are two challenges when applying a GA to an agent-based model: 1) how to design a fitness function that can consider the behaviors of all the agents; and 2) how to handle the stochasticity in the simulation run. We address the former by using Kullback–Leibler divergence [13] that measures the distance between two distributions; and the latter by exposing the random number seed in the model as a parameter to be calibrated by GA.

We test our hybrid approach on an agricultural agent-based model that we developed for the Monze District, Zambia, an area that is approximately 1,866 square miles in size, with 53,491 households. Each agent in the model represents a household with characteristic variables (e.g., number of household members and area of cultivation) and decision making rules (based on the variable values). Available data includes: 1) an extensive household survey conducted in Southern Province of Zambia, 2) district-level farmer registry data and 3) the Post Harvest Survey (PHS) data. The extensive survey data was collected through surveys of 330 households and includes information on household size from which labor supply and food demand are calculated. The farmer registry is compiled by regional agricultural extension officers and consists of a census of all small-scale farmers in a particular district with basic attributes, such as the total area of the farm and the total area under cultivation in a particular year. The Post Harvest Survey data is the mechanism that the Zambian government uses to assess end of season crop production (i.e. crop yield). The PHS is a household-level survey conducted with a sample of households in the country each year.

Agent-based models are highly sensitive to how the initial set of agents are created within a model (i.e. how many agents and with what attributes) and the decision algorithms that govern agent behavior in the model. It is rare to have an empirical dataset that tells a modeler exactly how many agents there should be in a model and what attribute are needed in order to represent a real-world scenario. In the case of our agricultural system in Zambia, we specifically need to know how many farmers there are in a particular area and how large their farms are in order to model farm-level agricultural production. No single dataset provides this information. Thus we have developed a hybrid data integration process drawing on the datasets above to initialize our farmer agents. Our hybrid draws on household attributes (household size) from our extensive survey data, land in cultivation from the farmer registry data and end of season agricultural yield from the Post-Harvest Survey data. From these three datasets, we are able to integrate these salient characteristics to create a set of farmer agents that enables us to run simulations for alternative climate scenarios.

We demonstrate that our approach is effective in integrating household survey data with the farmer register data, and in deriving an optimized set of values for agent variables based on PHS. It is our belief that the administrative register data provides the structure and heterogeneities of the real population at the individual level, which has never been done before to our best knowledge. The result shows that the simulated data can reflect the marginal distributions and the randomness from the survey data, and that the set of values optimized for the missing variables can produce simulated production close to the observed PHS.

The remainder of the paper is organized as follows. Section II reviews related work. Section III introduces the proposed simulating method and Section IV describes the application of microbial genetic algorithm to agent-based model. Section V demonstrates the experiment of proposed hybrid approach on the Zambia agent-based model. We conclude the paper in Section VI with future work.

II. Related Work

The existing work on creating household agents in ABMs for agricultural analysis [3], [4], urban planning [5] and urban disaster management [6] focused on decomposing aggregated demographic/administrative data. In environmental modeling, methods that create agents from survey data are often called parameterisation [14] and agent typology [15]. For example, Ralha et al. [16] use survey/sample data to create the agent typologies and then create a population of agents according to the distribution of agent types. In addition, there is research that maps disaggregated census data (like household level records) to various agent types [17], using techniques such as Regression Tree [18]. However, none of these methods leverages the relations from survey data to filling in missing variables in available real population data like our approach does. To our best knowledge, we are the first to integrate the real population data into the agent creation process.

Many agent-based models built in environmental science have a module that allocates land to agents. For example, Gaube et al. [19] and Ralha et al. [16] use a density/probability map to place households; Schouten et al. [20] and Murray-Rust et al. [21] introduce an auction/competition mechanism to allocate land to households. In our proposed method, we consider the spatial location an important variable of household agent and develop a land allocation algorithm that aims at forming natural farmer communities when placing the household agents.

Data imputation [22] focuses on missing variables inside a dataset. Our problem is different in that the administrative register data is a complete dataset, but it lacks some variables that are important to the agent-based simulations. Besides, most data imputation techniques work great when there is only a small percent of data missing, but in our case, the amount of missing data is two orders of magnitude larger than that of the known data. Lastly, data imputation is mostly followed by data analysis. Thus the design of data imputation techniques usually aims at supporting data analysis. However, in Agent Based Modeling (ABM), the set of agents is used to model the real world population.

Synthetic data generation and *data simulation* are terms usually referred to the creation of large population from a

small data – whether it is an aggregated dataset [2] and/or a sample data [23], [24], [25]. Data generation/simulation based on aggregated data usually means to decompose the data while maintaining the same marginal distribution on each dimension. Frazier and Alfons [26] utilize the linear regression model plus random errors, which is particularly similar to our method. However, instead of using aggregated data, our method leverages the large administrative register and the individual level survey data, which can better preserve the heterogeneity and the randomness.

Genetic Algorithms (GAs) can automatically search a parameter space, and thus they have been used to calibrate agent-based models [9], [8], [10], [11]. Calvez and Hutzler [9] summarize the specific difficulties related to the use of GA approach with agent-based models: the choice of fitness function, the stochasticity, and the computational cost. In this work, for agent variables that cannot be deducted from the available real world data, we apply the microbial GA [12] to find a set of values that can produce simulation outcome that is close to real world observations. We use the distance between the simulated outcome and real world observations as the fitness score; and we set the random number seed in the agent-based model and expose the random number seed as a parameter in the chromosome of GA to handle its stochasticity.

III. Simulation of Synthetic Population

This section introduces our proposed method that simulates the agent variables from individuals' register data, survey data, and remote sensing data.

A. Simulating Household Characteristics from Survey Data and Real Population Data

The first step in the data simulation is to create a generalized linear model based on the relationship between a set of observed independent variables and an observed response variable in the survey data – while the same independent variables are known in the register data, the response variable is a missing variable in the register data. Generalized linear models [27] are extensions of traditional regression models that allow the mean to depend on the independent variables through a link function, and the dependent variable to be any member of a set of distributions called the exponential family (e.g., Normal, Poisson, Binomial). By using the generalized linear model we make two assumptions:

1) That there exists a strong correlation between the independent variables and the dependent variable(s) . This can be tested with Likelihood Ratio Test (LRT) on a fitted model.
2) That the dependent variable(s) can be modeled using one of the distributions in the exponential family. This can be verified using a goodness of fit test.

For instance, when there is only one independent variable (X) and the dependent variable has a Poisson distribution, the fitted model can be represented as:

$$log(E(Y|X)) = a + b * X \tag{1}$$

where Y denotes the dependent variable, $E(Y|X)$ denotes the expected value of Y given X, and a, b are the coefficients estimated during the model fitting process.

Once the generalized model is fitted, we apply it to the independent variables in the farmer register to predict the mean values of the missing variables. We then use the predicted mean value to randomly create values based on the distribution in the exponential family that we chose.

B. Simulating Household Spatial Locations by Allocating Agricultural Land to Household Agents

Spatial interactions amongst agents require the exact spatial locations of agents to be known. However, the spatial information contained in the real population data is generally given in aggregated form (*e.g.,* by zones). To solve that, we take the remote sensing data that is classified into agricultural and non-agricultural land, disaggregate it into raster, and allocate the agricultural cells/patches to households.

We developed an algorithm shown in Fig. 1, to allocate the agricultural cells to households. Our land allocation algorithm first chooses a number of seed households in the procedure ALLOCATE_LAND_TO_HH. For each seed household H, it randomly selects an unallocated farmland patch SP and then invokes ALLOCATE_MANY that assigns to SH with P and extra unallocated farmland patch(es) within the maximum search radius s of P. Within ALLOCATE_MANY, ALLOCATE_ONE is invoked to actually allocate one farmland to a household, and mark the adjacent farmland as $tentative$. Once all seed households are assigned with enough farmland, ALLOCATE_LAND_TO_HH will continue to allocate farmland to the non-seed households. For each non-seed household SH, it finds a $tentative$ unallocated farmland TSP and invoke ALLOCATE_MANY(SH, TSP). In this way, the households should be located close to each other to form communities.

IV. Calibrating Agent Variables with Genetic Algorithm

Genetic Algorithm (GA) is a heuristic search that mimics the process of natural selection. It belongs to the larger class of evolutionary algorithms (EAs), which generate solutions to optimization problems using techniques inspired by natural evolution. While there are different variants of GA, the common underlying idea is the same: given a population of individuals and a fitness function, the properties of the individuals are mutated and altered in each generation and the best fitted individuals are preserved to the next generation to evolve an optimized solution. The Microbial Genetic Algorithm is a minimal GA that has the same functionality and efficacy as the standard GAs, but is simple to code and tune. We choose to implement the Microbial GA instead of using other off-the-shelf software for the simplicity and a better understanding of the calibration process.

The most creative and challenging parts of programming a GA are usually the problem-specific aspects, that is, the design of chromosome (a set of properties for each individual in the population) and its mutation/alternation process, and the fitness

```
1: procedure ALLOCATE_ONE(H, P)  ▷ H: household, P: patch
2:     A ← area of farmland needed by H
3:     if A > area of P then
4:         occupiedRatio(P) ← 1 ▷ P fully occupied by H
5:     else
6:         occupiedRatio(P) ← (A − 1)  ▷ P partially occupied by H
7:     end if
8:     N ← neighbor farmland (in radius r) of P  ▷ r a global parameter of allocation radius
9:     status(N) ← tentative seed patches
10: end procedure

11: procedure ALLOCATE_MANY(H, P)  ▷ H: household, P: patch
12:     Invoke allocate_one(H, P)
13:     repeat
14:         searchRaius ← 700m  ▷ starting from threshold value
15:         SP ← randomly selected unoccupied farmland within searchRaius of P
16:         if SP is not NULL then
17:             Invoke allocate_one(H, SP)
18:         else
19:             searchRaius ← (searchRaius + 100m)  ▷ Expand the search area
20:         end if
21:     until H has been assigned enough farmland ∨ searchRaius == s  ▷ s is global parameter of maximum search radius
22: end procedure

23: procedure ALLOCATE_LAND_TO_HH
24:     i ← 1  ▷ id of current HH to be allocated
25:     repeat
26:         SH ← the ith household
27:         status(SH) ← seed household
28:         SP ← a randomly selected patch
29:         Invoke allocate_many(SH, SP)
30:         i ← (i + 1)
31:     until i == numSeed ∨ no unoccupied land  ▷ numSeed is global parameter of total number of seed households created during initialization
32:     repeat
33:         SH ← ith household
34:         TSP ← randomly selected patch so that status(TSP) == tentative seed patch ∧ occupiedRatio(TSP) == 0
35:         Invoke allocate_many(SH, TSP)
36:         i ← (i + 1)
37:     until i == numHHs ∨ there is no unoccupied land  ▷ numHHs is total number of households
38: end procedure
```

Fig. 1. Algorithm to allocate agricultural cells to households. Comments are denoted by right-pointing triangle.

function (the fitness score is usually the objective value in the optimization problem being solved).

The chromosome could be composed of properties that each represents a missing agent variable. There are different types of missing variables and thus each has to be treated differently:

1) For nominal variables, such as $soilType$, we represent them as integers, and mutate them randomly into any other possible values.
2) For continuous variables, such as $ratioOfLocalMaize$, we represent them as doubles, and mutate them using a Gaussian number generator.
3) For variables that follow a certain distribution, we expose its parameters as doubles and mutate them using a Gaussian number generator. For example, we assume that $plantingDate$ follows a normal distribution and choose to fix its mean while changing its standard deviation.

Note that the mutation/alternation has to be within the value space of each variable, and if it results in a value that falls below or goes above the boundaries, we simply replace it with the boundary values.

Two simulation runs of the same agent-based model can generally bring slightly different results, due to the stochasticity of the model and the simulator. One solution is to simulate each model several times to improve the evaluation of the fitness function, which however greatly increases the number of simulations and thus the time to run the simulations. Calvez and Hutzler [9] address this issue by estimating the fitness of each model with one simulation at most generations, while simulate the model several times at each n generation. While their method reduces the number of simulations, it still needs to simulate one model several times. To completely avoid it, we propose to fix the random number seed in the agent-based model (and the platform) and expose it as a parameter to the genetic algorithm. In this way, the model becomes deterministic while the genetic algorithm can factor in the stochasticity of the model when searching for the best combination of parameters.

Lastly, agent-based systems or simulations are dynamic and often characterized by emergent phenomena, which complicates the measure of the fitness function. What has to be measured strongly influences the characteristics of the models obtained by the genetic algorithm. If the fitness function is not carefully chosen, the resulting models will be optimized for that specific fitness function. Since the data generated from agent-based model can be collected at individual level (e.g., the yield of each household agent) or aggregated level (e.g., total crop production), we could evaluate the fitness based on the individual level data or the aggregated data. For example, we can measure the distance between the simulated average yield of household agents and the observed average yield from PHS (aggregate level); or we can measure the difference between the distribution of simulated household yields and that of observed household yields in PHS (individual level). In this paper, we propose to use the Kullback–Leibler divergence to measure the difference between the distribution of simulated

data and the distribution of observed data. This is because we want the agents' behaviors to accurately reflect the variance in real households' crop production.

V. Application to Zambia Food Security ABM

A. Zambia Crop Decision-making ABM

We apply our technique to the initialization of a spatially explicit ABM of intra-seasonal agricultural decision-making of households for the entire Monze District in Zambia. In our model, household agents make decisions biweekly based on a utility maximization approach [5] within the context of local institutional regimes (i.e., wards). The goal is to use this model to identify how climate change impacts adaptive capacity.

B. Survey Data and Farmer Register Data Cleaning

The original survey data and farmer register data are stored in Excel spreadsheet. We start with writing Python code to auto-correct the formatting errors and typos, and to extract the information into a MySQL database. However, additional cleansing has to be done to prepare the data for simulation. Cultivated area ($CultArea$) is expressed in hectares out to two decimal places in the survey data, but is frequently rounded in the farmer register. The rounding could be because people tend to answer with rounded values, which is called *response heaping* and has been extensively observed and studied [28]. To address this problem, we decided to round the variable of $CultArea$ in both the survey data and the farmer register data. The rounding policy is described below:

1) If the value of $CultArea$ is less than 1ha, we round it up to 1ha. This is to avoid rounding small values of $CultArea$ to zero, since the data is collected from smallholder farmers (that each owns small-based plots of land).
2) For any other value of $CultArea$, we round it to the nearest integer value.

Incorrect records can exist in both the survey data and the register data. We identified and removed one record (out of 184) in the survey data with a $CultArea$ of 80ha, which is far larger than the others (maximum 12ha, see Figure 2). From the register data, we removed records that have $CultArea$ larger than $FarmArea$ – the total area owned by the farmer's household. Since the survey data after cleaning has $CultArea$ ranging from 0ha to 12ha, and there are only 107 records (out of 53597) in the register data that have $CultArea$ larger than 12ha, we decided to remove the 107 records from the register data. This makes sure that the generalized linear model trained from survey data is applied to the same range of independent variables in the register data.

Another concern with the survey data is whether it is an actual random draw from the population data. To test that, we perform a two-sample Kolmogorov-Smirnov test on the rounded variables of $CultArea$ using the function 'ks.test' in R [29]. The result gives a p-value of 0.1629, which suggests that the probability distributions of rounded $CultArea$ in the survey data and in the register data are consistent – the reason for this p-value being not significant might be the response

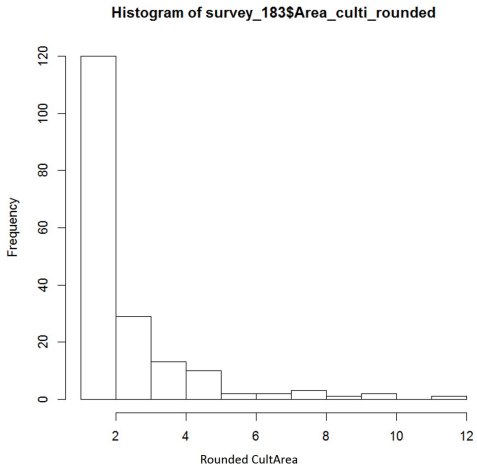

Fig. 2. Histogram of the rounded values of 'Area_culti' in the survey data.

heaping problem that we mentioned before. Fig. 3 shows that they have similar Empirical Cumulative Distribution Functions (ECDFs).

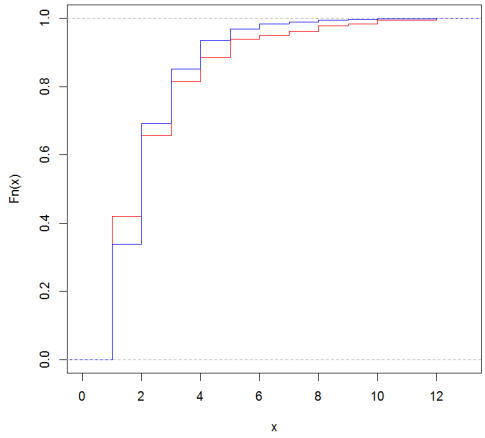

Fig. 3. Comparison of the ECDFs (red: the rounded variable $CultArea$ from survey data; blue: the rounded variable of $CultArea$ from the register data).

C. Remote Sensing Data

The Monze land cover dataset was generated from the spectral classification of medium-resolution Landsat 5 Thematic Mapper (TM) data. Multi-temporal data were utilized in the analysis, with scenes acquired on September 29, 2008 and May 24, 2009 (nominal scene center of Path: 072 and Row: 071). Land cover within the study area was classified into five primary categories: forest, cropland, savanna, settlement/urban, and water with an emphasis on the accurate delineation of cropland.

An ISODATA algorithm was initially applied to the multi-temporal data to segment pixels into natural clusters reflecting the underlying structure of the data. Clusters were randomly sampled and sample locations were coded with the appropriate land cover label using high resolution imagery. Spectral signatures were extracted at sample locations and statistically

clustered into primary category subgroups, ranging from 3-10 subgroups for non-water categories. Subgroup signatures were then used to parameterize a maximum likelihood classifier.

An accuracy assessment of the subgroup thematic map indicated which subgroups, or strata, of primary categories exhibited high commission error. Logit models paired with multi-spectral derivatives were adopted to correct class confusion within high-error strata. Pixels within targeted strata were reclassified based on the predicted probabilities of membership to a particular primary category. In total, seven strata directly affected the accuracy of cropland and were reclassified. Overall accuracy for the five primary categories was 88.18%. Reclassification of select strata reduced the spatial extent of the initial cropland class by over 53% and reduced error of omission to 12.1% and error of commission to 9.8%.

D. Household Characteristics Simulation

The household size (variable $HHSize$ in the survey data) is the number of members in a household. This integer data can be modeled as a Poisson distribution as determined by a goodness-of-fit test (function $goodfit$ in R) on the values of $HHSize$ in the survey data (p-value 0.056). See Fig. 4 for the histogram and a rootogram of the observed and fitted values.

We then used a generalized linear model (specifically glm method in R) to fit using the variables $CultArea$ (rounded) and $HHSize$ from the survey data.

$$Log(E(HHSize|CultArea)) = 1.70118 + 0.06279 * CultArea \quad (2)$$

The fitted model can be visualized with the survey data (Fig. 5).

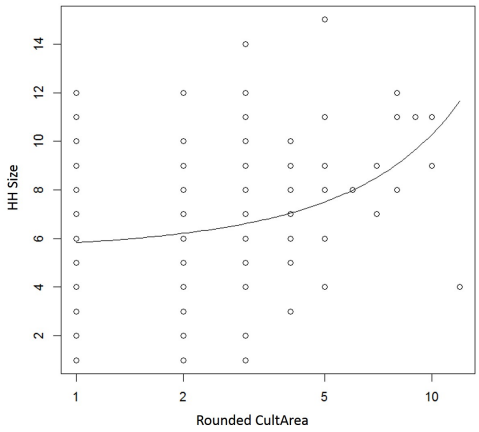

Fig. 5. Visualization of the fitted model using the survey data. Note that the X-axis is in log scale.

Using R's $summary$ method on the fitted model we can see the results of Likelihood Ratio Test (LRT). LRT compares the fitted full model with the restricted model where the independent variable $CultArea$ is omitted. The p-value of 1.98e-06 suggests that the dependent variable $HHSize$ is strongly related to the independent variable $CultArea$. Note that while there might be other predictors of $HHSize$ and the generalized model can have more than one predictor, $CultArea$ is the only predictor we have in our input data, and our goal is to simulate the missing variables as best as we can.

To simulate the values of $HHSize$ in the register data, we first used the fitted model to predict the value of $HHSize$ for each value of $CultArea$ in the register data. Then we used the predicted values as the parameter $lambda$ (mean value) in Poisson distribution to randomly generate the simulated values of $CultArea$ (here we use R's $rpois$ method). The simulated data can be plotted together with the survey data (Fig. 6). It can be seen that the simulated data points (blue) appear like "expanding" from the original survey data (red).

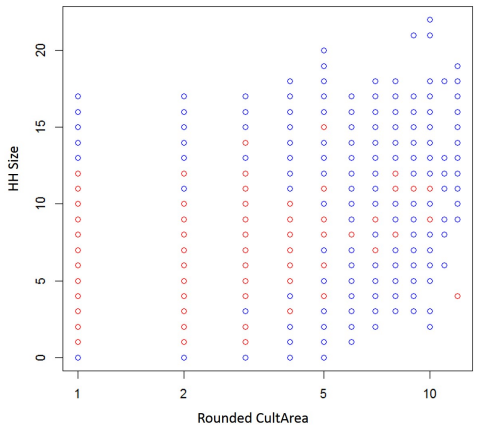

Fig. 6. Overlaid visualization of the simulated data (blue) and the survey data (red). Note that the X-axis is in log scale.

To verify that the simulated data has the same marginal distribution as the survey data, we run the two-sample Kolmogorov-Smirnov test. The resulting p-value of 0.999 shows that the two do have the same probability distribution. This can also be seen from the overlaid Empirical Cumulative Distribution Function (ECDF) in Fig. 7.

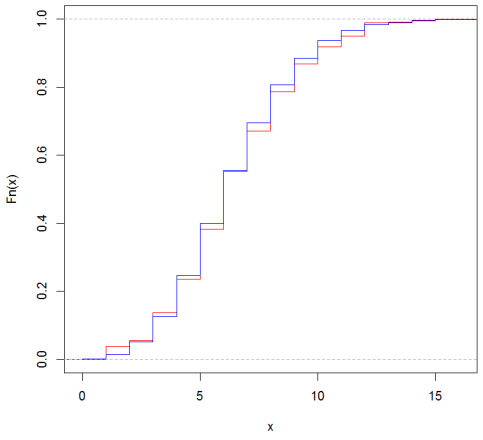

Fig. 7. Comparison of the ECDFs of the variable HHSize (red: survey data; blue: simulated data).

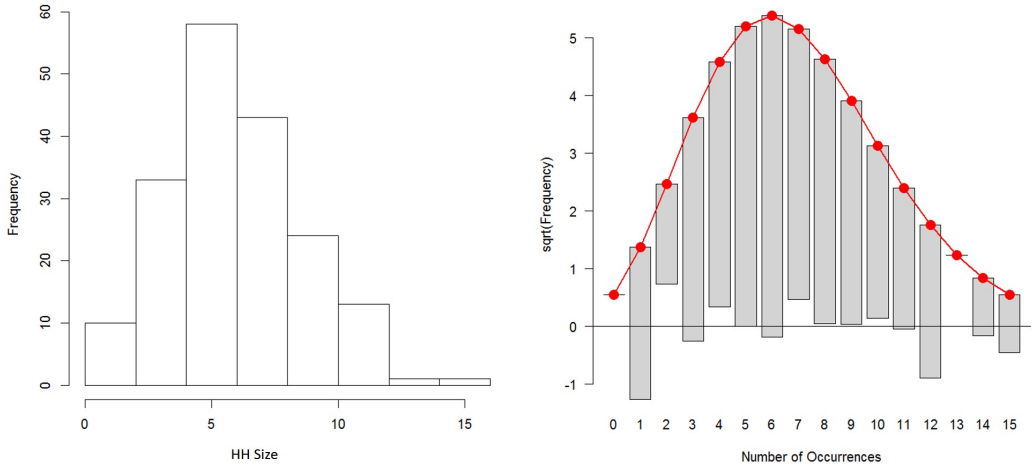

Fig. 4. On the left is a histogram of variable $HHSize$ in the survey data and on right is rootogram of the observed and fitted values.

However, it is not expected for the simulated data to have the same multivariate distribution as the survey data, as the register data has a different distribution of values of $CultArea$. This can be shown by overlaid visualization of the kernel density estimates (Fig. 8). We also ran a kernel density comparison test on those two distributions to confirm that they are different (the R function $kde.test$ generates a p-value of 0).

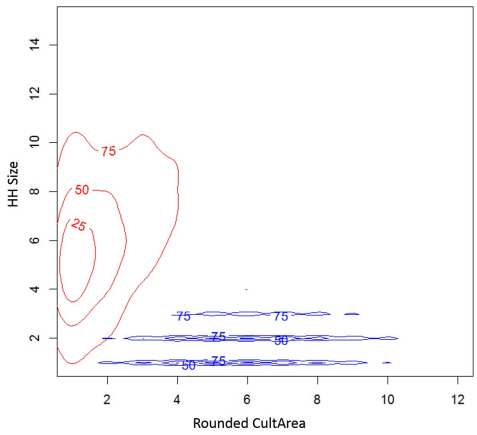

Fig. 8. Overlaid visualization of the kernel density estimates for the simulated data (blue) and the survey data (red).

E. Household Spatial Location Simulation

Next, we add the exact spatial locations to the simulated population of households by allocating to them the agricultural cells/patches generated from remote sensing data. The register data has the aggregated spatial information that tells the ward name of each household. We break down the entire land cover raster of Monze District based on wards, and then run the land allocation algorithm for each ward. Fig. 9 shows the results for one ward.

Fig. 9. Results of land allocation in one ward of Monze District, Zambia. Left: agricultural land (brown) and non-agricultural land (green); Right: agricultural land is allocated to households (red).

F. Remaining Missing Variables Calibrated by Microbial GA

Finally, we calibrate all the missing variables whose values could not be determined in previous steps, using the Microbial Genetic Algorithm. Each chromosome is composed of four properties: $soilType$ (integer, 0–14), $ratioOfLocalMaize$ (double, 0–1), $plantingDateStandardDeviation$ (double, 0.001–0.167), and $randomSeed$ (integer). Among those properties, $soilType$ and $ratioOfLocalMaize$ are direct representations of agent variables — there are 15 types of soil in the model, and there are two types of maize (hybrid or local). The possible planting dates within a growing season are every other weeks from middle October to end of January — eight options in total. Finally, we assume that the discrete probability of planting dates follows a normal distribution, that is, we assume most data points fall into 0–1 on the x-axis which is equally divided into eight intervals (eight options

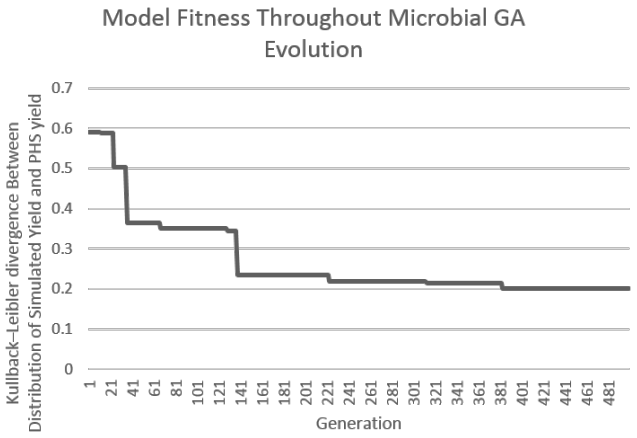

Fig. 10. Model fitness throughout Microbial GA evolution with a population size of 50 and a deme size of 25.

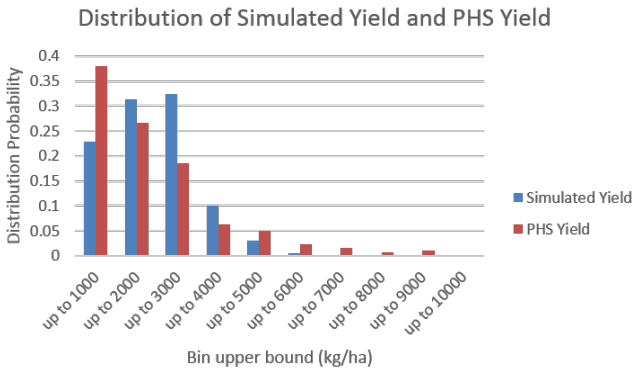

Fig. 11. The comparison between simulated yield distribution and the observed yield distribution from PHS.

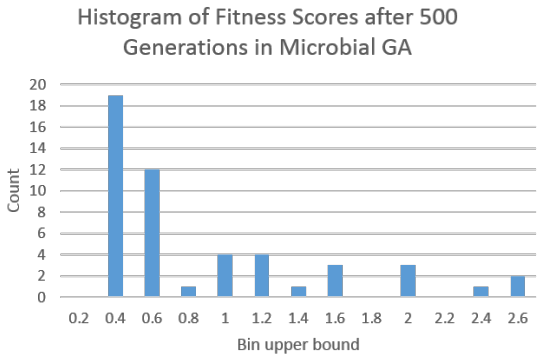

Fig. 12. The distribution of fitness scores (KullbackLeibler divergence) after 500 generations in Microbial Genetic Algorithm.

by the temporal order), and use the area of distribution on each interval as the probability of planting dates falling into that option. We fix the mean value to be the middle of the planting date options, and expose its standard deviation as $plantingDateStandardDeviation$. To ensure that most data points in the normal distribution fall between 0–1, we limit the maximum standard deviation to 0.167 (the approximation of $0.5/3$). Thus $plantingDateStandardDeviation$ ranges from 0.001 (which has to be larger than zero) to 0.167 (both are inclusive).

We randomly created a population of 50 chromosomes, and evaluated the fitness of each chromosome by running the simulation and measure the Kullback–Leibler divergence between the distribution of simulated yield and that of the observed yield in PHS, for growing season 2011–2012. We calculated the fitness score for each chromosome and stored it into a cache table and updated the score only when that chromosome is mutated/altered, so that each time when a comparison was made between two chromosomes, we can look up the cache table to avoid recalculating the fitness scores.

The deme size in the microbial GA is used to maintain a trivial geography. Spector and Klein [30] chose the deme size arbitrarily and claim that "trivial geography will often provide benefits with a range of deme size and that the choice of deme size is not critical." In our experiment, we set the deme size to be 50% of the population size, which is 25, and run the Microbial GA for 500 generations. Figure 10 shows that the fitness score (Kullback–Leibler divergence) decreases as the population evolves in the Microbial GA.

Figure 11 shows that the simulated yield has a distribution similar to that of the observed yield from PHS.

Figure 12 shows the distribution of fitness scores, and it can be seen that the entire population tends to converge to the optimal (minimal) fitness score. What is more interesting is to see if there are different sets of values that can all produce close to optimal fitness scores. Table I shows the details of the chromosomes that fall into the first bin (0.2–0.4). It can be seen that:

1) There are several different $soilType$ that can produce the optimal results;
2) $ratioOfLocalMaize$ tends to converge to 0.57;
3) $plantingDateStandardDeviation$ converged to 0.167;
4) The random number seed can be very different.

To summarize, we applied the Microbial GA to calibrate the missing variables that cannot be derived in previous steps, and the result shows good matching in distributions of simulated data and observed data.

G. Discussion

An agent in ABM is generally characterized by lots of parameters which together determine the global dynamics of the model. Thus when calibrating these parameters, the search space is huge and Genetic Algorithms (GAs) are good at dealing with the large dimensionality of this problem. There is no mathematical equation that can anticipate the dynamics of an agent-based model without executing it, and thus the computation of the fitness function requires the execution of simulation, which implies a high computational cost. Although more computationally intensive than traditional nonlinear es-

TABLE I
THE SETS OF PARAMETERS THAT CAN PRODUCE THE BEST FITNESS SCORES.

score	$soilType$	$ratioOfLocalMaize$	$plantingDateStandardDeviation$	$randomSeed$
0.20116529	WI_LVLS007	0.572238345	0.167	-6.53E+08
0.214395283	WI_LVLS007	0.504317117	0.167	-1.45E+09
0.218892992	WI_GLBW752	0.598754994	0.167	1.73E+09
0.218977268	WI_GLBW752	0.598754994	0.167	2.04E+09
0.224874894	WI_GLBW752	0.687669638	0.167	7.33E+08
0.227640214	WI_GLBW752	0.557108806	0.167	9.41E+08
0.236832655	WI_GLBW752	0.598754994	0.161571945	9.29E+08
0.247164101	WI_LVLS007	0.879742596	0.167	-4.08E+08
0.272959905	WI_LVLS007	0.926399603	0.167	-1.41E+09
0.293541824	WI_GLBW752	0.879742596	0.167	-4.61E+08
0.299738564	WI_GLBW752	0.879742596	0.167	-7.63E+08
0.317545137	WI_LVLS007	1	0.167	1.62E+09
0.323698096	WI_LVLS007	1	0.167	4.99E+07
0.326112368	WI_LVLS007	1	0.167	-6.53E+08
0.326112368	WI_LVLS007	1	0.167	-6.53E+08
0.329442109	WI_LVLS007	1	0.167	1.92E+09
0.344025581	WI_PHCF014	1	0.167	-5.85E+08
0.351703556	WI_CMZR003	1	0.167	8.21E+08
0.358091846	WI_PHCF014	1	0.167	1.62E+09

timation techniques, the GA is capable of accurately finding optimum parameter sets and providing additional information about the search space (Table I). While Monte Carlo experiments could be conducted to generate similar information about the search space, GA is much more efficient as it is integrated with the optimization process. In addition to the computational cost, two other difficulties of using GA in ABM are the choice of fitness function and the stochasticity of simulation run. In Section V-F, we demonstrate that our proposed method can successfully address these two problems.

VI. CONCLUSION AND FUTURE WORK

In this paper we propose a hybrid method that can create a synthetic population to reflect the structure and heterogeneities of real farmer households, and to optimize the *replicative validity* of the model matching data already acquired from the real system (retrodiction). While existing research generally does not use real population data directly in creating synthetic population, due to its lack of information, we demonstrate its applicability in agent-based modeling by integrating other data sources with the real population data. For agent variables whose values cannot be determined due to lack of data, we propose to use Genetic Algorithm (e.g., the Microbial GA) to search for a set of values that can match the model to production survey data (e.g., PHS). We expose the random number seed of the model and the platform as a property of the chromosome to be determine by GA, and we evaluate the fitness based on the distribution of simulated yield using the Kullback–Leibler divergence. We have applied the method to our food security agent-based model in Zambia. The result shows that the synthetic population generated using our method can reflect the marginal distributions of aggregated survey data, and the distribution of simulated yield is close to that of the observed yield.

The next step is to use the synthetic population as the basis and continue developing the Zambia agent-based model to study the interactions between household agents (e.g., labor sharing) and the impact of climate change on food security. We will evaluate whether or not the generated synthetic population can achieve good *predictive validity* – the model matches data before data is acquired from the real system. Finally, there are other parameter search methods such as Reinforcement Learning, and we will compare them with the GA method.

ACKNOWLEDGMENT

The authors thank Sean Sweeney at Indiana University for generating the landcover dataset from the remote sensing data. The research is supported in part by the National Science Foundation under grants BCS1026776 and SES-1360463, and by the Pervasive Technology Institute at Indiana University.

REFERENCES

[1] T. Berger, "Agent-based spatial models applied to agriculture: a simulation tool for technology diffusion, resource use changes and policy analysis," *Agricultural economics*, vol. 25, no. 2, pp. 245–260, 2001.

[2] R. Moeckel, K. Spiekermann, and M. Wegener, "Creating a synthetic population," in *Proceedings of the 8th International Conference on Computers in Urban Planning and Urban Management (CUPUM)*, 2003.

[3] T. P. Evans and H. Kelley, "Multi-scale analysis of a household level agent-based model of landcover change," *Journal of Environmental Management*, vol. 72, no. 1, pp. 57–72, 2004.

[4] H. Kelley and T. Evans, "The relative influences of land-owner and landscape heterogeneity in an agent-based model of land-use," *Ecological Economics*, vol. 70, no. 6, pp. 1075–1087, 2011.

[5] R. J. Beckman, K. A. Baggerly, and M. D. McKay, "Creating synthetic baseline populations," *Transportation Research Part A: Policy and Practice*, vol. 30, no. 6, pp. 415–429, 1996.

[6] D. Felsenstein, A. Y. Grinberger, and M. Lichter, "Dynamic agent based simulation of an urban disaster using synthetic big data," in *Workshop on Big Data and Urban Informatics*, 2014.

[7] K. G. Troitzsch, "Validating simulation models," in *18th European Simulation Multiconference. Networked Simulations and Simulation Networks*, 2004, pp. 265–270.

[8] O. B. Espinosa, "A genetic algorithm for the calibration of a micro-simulation model," *arXiv:1201.3456*, 2012.

[9] B. Calvez and G. Hutzler, "Automatic tuning of agent-based models using genetic algorithms," in *Multi-agent-based simulation VI*. Springer, 2005, pp. 41–57.

[10] Z. Y. Wu, T. Walski, R. Mankowski, G. Herrin, R. Gurrieri, and M. Tryby, "Calibrating water distribution model via genetic algorithms," *Proc. AWWA IMTech, Kansas City, Mo*, 2002.

[11] A. E. Mulligan and L. C. Brown, "Genetic algorithms for calibrating water quality models," *Journal of environmental engineering*, vol. 124, no. 3, pp. 202–211, 1998.

[12] I. Harvey, "The microbial genetic algorithm," in *Advances in artificial life. Darwin Meets von Neumann*. Springer, 2009, pp. 126–133.

[13] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[14] T. Iwamura, E. F. Lambin, K. M. Silvius, J. B. Luzar, and J. M. Fragoso, "Agent-based modeling of hunting and subsistence agriculture on indigenous lands: Understanding interactions between social and ecological systems," *Environmental Modelling & Software*, vol. 58, pp. 109–127, 2014.

[15] D. Valbuena, P. H. Verburg, and A. K. Bregt, "A method to define a typology for agent-based analysis in regional land-use research," *Agriculture, Ecosystems & Environment*, vol. 128, no. 1, pp. 27–36, 2008.

[16] C. G. Ralha, C. G. Abreu, C. G. Coelho, A. Zaghetto, B. Macchiavello, and R. B. Machado, "A multi-agent model system for land-use change simulation," *Environmental Modelling & Software*, vol. 42, pp. 30–46, 2013.

[17] A. Smajgl, D. G. Brown, D. Valbuena, and M. G. Huigen, "Empirical characterisation of agent behaviours in socio-ecological systems," *Environmental Modelling & Software*, vol. 26, no. 7, pp. 837–844, 2011.

[18] A. Smajgl and E. Bohensky, "Behaviour and space in agent-based modelling: Poverty patterns in east kalimantan, indonesia," *Environmental Modelling & Software*, vol. 45, pp. 8–14, 2013.

[19] V. Gaube and A. Remesch, "Impact of urban planning on household's residential decisions: An agent-based simulation model for vienna," *Environmental Modelling & Software*, vol. 45, pp. 92–103, 2013.

[20] M. Schouten, T. Verwaart, and W. Heijman, "Comparing two sensitivity analysis approaches for two scenarios with a spatially explicit rural agent-based model," *Environmental Modelling & Software*, vol. 54, pp. 196–210, 2014.

[21] D. Murray-Rust, C. Brown, J. Van Vliet, S. Alam, D. Robinson, P. Verburg, and M. Rounsevell, "Combining agent functional types, capitals and services to model land use dynamics," *Environmental Modelling & Software*, vol. 59, pp. 187–201, 2014.

[22] T. D. Pigott, "A review of methods for missing data," *Educational research and evaluation*, vol. 7, no. 4, pp. 353–383, 2001.

[23] A. Alfons, S. Kraft, M. Templ, and P. Filzmoser, "Simulation of synthetic population data for household surveys with application to eu-silc," Research Report CS-2010-1, Department of Statistics and Probability Theory, Vienna University of Technology. URL http://www. statistik. tuwien. ac. at/forschung/CS/CS-2010-1complete. pdf, Tech. Rep., 2010.

[24] T. Arentze, H. Timmermans, and F. Hofman, "Creating synthetic populations: approach and empirical results," in *Proceedings of the AET European Transport Conference*, 2001.

[25] ——, "Creating synthetic household populations: problems and approach," *Transportation Research Record: Journal of the Transportation Research Board*, 2015.

[26] T. J. Frazier and A. Alfons, "Generating a close-to-reality synthetic population of ghana," *Social Science Research Network (SSRN) 2086345*, 2012.

[27] P. McCullagh, "Generalized linear models," *European Journal of Operational Research*, vol. 16, no. 3, pp. 285–292, 1984.

[28] A. L. Holbrook, S. Anand, T. P. Johnson, Y. I. Cho, S. Shavitt, N. Chávez, and S. Weiner, "Response heaping in interviewer-administered surveys is it really a form of satisficing?" *Public Opinion Quarterly*, vol. 78, no. 3, pp. 591–633, 2014.

[29] R. Ihaka and R. Gentleman, "R: a language for data analysis and graphics," *Journal of computational and graphical statistics*, vol. 5, no. 3, pp. 299–314, 1996.

[30] L. Spector and J. Klein, "Trivial geography in genetic programming," in *Genetic programming theory and practice III*. Springer, 2006, pp. 109–123.